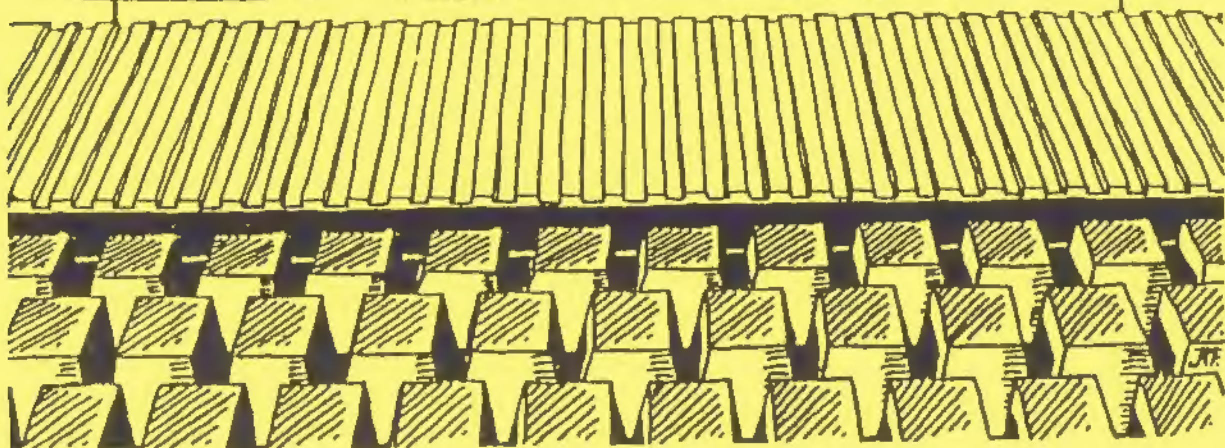
Atom Nieuws

JAARGANG : 15
NUMMER : 1
DISK 96-1



FEDERATIE VAN ATOMCLUBS NEDERLAND - BELGIE.

Voorzitter :	Secretaris:	Penningmeester:
P.v.Kuik	J.Hartog	T.Rutten
Zuideinde 54-a	Keyenbergseweg 60	Berkenlaan 24
1843 JP Groot-Schermer	6871 WK Renkum	3737 RN Groenekan
tel. 0299-67.19.02	tel. 0317-31.37.57	tel.0346-21.34.95

Contributie 1996 : fl. 25,00 : Atom Computerclub ; Giro 5244293.

Redactie Atom Nieuws	Redactieadres A.N.	Ledenadministratie
B.Tossaint 043-343.16.75	B.Tossaint	T.Rutten
W.Truijen 00-3289564792	Fatimaplein 85	Berkenlaan 24
R.Leurs 0345-57.29.70	6214 TW Maastricht	3737 RN Groenekan
	tel. 043-343.16.75	tel.0346-21.34.95

UITERSTE DATUM INLEVERING KOPY VOOR NR. 15-2 : 1 JUNI 1996

Clubwinkel	ATOM-BULLETIN-BORD speciaal v. ATOM-in-PC	
J.Hartog	R.Bronsdijk	Unicorn BBS (H.Derksen)
Keyenbergseweg 60		026-422.55.06 in de gebieden
6871 WK Renkum	inloggen op	ATOMFILE en ATOMMESSAGE de
tel.0317-31.37.57	tel.020-651.28.16	laatste versies ATOM-PC -
		systeemsoftware .

VAN DE REDACTIE .

Even leek het erop dat dit een Roland-nummer zou worden . Maar toen kwam er toch nog een seintje van Leendert en werd ikzelf geconfronteerd met een aantal oude programma's om een man met beperkt gezichts-vermogen aan bezigheden te helpen. En, beginnen we dus welgemoed aan ons derde lustrum !!. Voor de toekomstvisie , zie pagina 22 .

Mede naar aanleiding van een hartekreet van Henri Derksen, deze keer weer eens een ledenlijst met de nieuwe telefoonnrs.erbij.

Namens de redactie
Bruno Tossaint

N.B. De inhoud van de regio-schijf : volgende uitgave

I N H O U D S O P G A V E

A.C.C.-LAND-DAG VOORJAAR 1996

DATUM : 27 APRIL 1996 ,van ca. 10.00 tot ca.17.00 u.
PLAATS : PAROCHIECENTRUM ,Melkweg 5,DE BILT.

Op de A27 de afslag Utrecht-Oost,maar let vooral op de kleine witte aanduiding =VEEMARKT=, de Biltse Rading afrijden tot de splitsing Groenekan/Bilthoven ,hier r.a. ,20 meter verder l.a.
Voor busreizigers : Vanaf Utrecht CS lijn 57,stopt op de Melkweg

DEMO's: o.a.* R.Leurs : ATOM IN PC, verbeterde versie
* L.Bijnagte : ATOM-in-PC onder Windows
Willy Truijen aanwezig met diskarchief, gal's en Eprom en wat verder ter tafel komt

Tevens Jaarvergadering Federatie van Atomclubs Nederland-Belgie

Het Bestuur van de Federatie.

TRUE-CPU

ofwel de waarheid omtrent OSBYTE #08

Geheel terecht wees Sjaak Geene me op de vorige landdag dat OSBYTE#08 niet altijd de juiste waarde retour geeft. Dat komt doordat OSBYTE#08 niet echt bepaalt welke processor in het systeem aanwezig is. Ergens in het operating systeem is opgeslagen welke processor in onze Atom draait en op welke snelheid. De oorspronkelijke bedoeling was dat, voordat de eprom geprogrammeerd werd, daar de juiste processorinformatie ingezet wordt. Dat is in de meeste gevallen nooit gebeurd, grotendeels omdat velen hun operating system eprom direct door Willy Truyen hebben laten klonen. Daarmee meldt bijna iedere OSBYTE#08 in Nederland welke processor in Willy's Atom draait.

Inmiddels hebben Leendert Bijnagte en ik een aantal routines gedemonstreerd om te bepalen welke processor werkelijk in uw systeem draait. Een van deze routines heb ik uitgebreid om ook de kloksnelheid te meten.

Deze snelheid wordt berekend door gebruik te maken van een externe tijdbasis. Met behulp van pc-commando #19 (geluid maken) wordt de PC voor 2/18e seconde geblokkeerd voor de Atom. De Atom laat een geluid produceren met frequentie 0 (niet om aan te horen) en stuurt direct daarna een byte naar de pc. Zolang de pc bezig is met het maken van het geluid wordt dit laatst verzonden byte niet gelezen. De Atom hangt dan niet vast, maar verhoogt ondertussen steeds een teller. Afhankelijk van Atoompjes klokfrequentie bereikt de teller een hoge of een lage eindstand. Hierdoor weten we dus wat de snelheid is.

```
 10 REM TRUE CPU
 20 DIM LL9,OO9,VV9
 30 FOR A=0 TO 9
 40  LL(A)=#FFF;OO(A)=#FFF;VV(A)=#FFF
 50 NEXT A;PRINT $21
 60 FOR A=0 TO 1
 70 P=#B00
 80[
 90:LL0 LDA @11;JSR #FFC2;BCC LL0
 95 STX #E4;STY #E5;PHP
 100 LDA @#00;JSR #FFC8;LDA @#19;JSR #FFC8
 110 LDA @#00;JSR #FFC8;LDA @2;JSR #FFC8
 120 LDA @#00;JSR #FFC8;LDA @#19;JSR #FFC8
 130 LDA @#00;JSR #FFC8;LDA @2;JSR #FFC8
 140 LDX @#00;TXA;TAY;STA #B810
 150:LL1 LDA #B81D;AND @#10;BNE LL2
 160 INX;BNE LL1;INY;JMP LL1
 170:LL2 LDA @#00;JSR #FFC8
 180 LDA @10;JSR #FFC2
 190 LDA @0;CPY @45;BMI LL3;ORA @#02
 200:LL3 PHA;SED;LDA @#99;CLC;ADC @#01;BMI LL4
 210 PLA;ORA @#01;JMP LL5
 220:LL4 PLA
 222:LL5 LDY #E5;LDX #E4;PLP
 224:LL6 RTS
 230:OO4 CMP @#08;BEQ LL0;JMP (VV0)
 260:OO0 LDA #220;STA VV0;LDA #221;STA VV0+1
 270 LDA @OO4%256;STA #220;LDA @OO4/256;STA #221;JSR OO6
 280 JMP (VV1)
 290:OO2 LDA #4FA;CMP @#4C;BNE OO3
 300 LDA #4FB;STA VV1;LDA #4FC;STA
```

```
VV1+1
 310:OO3 LDA @OO0%256;STA #4FB;LDA @OO0/256;STA #4FC
 320 LDA @#4C;STA #4FA
 330:OO6 JSR LL0;PHA;JSR #F7D1;];!P=#0D0A0B11;P=P+4
 340 $P="Processor: 65";P=P+LEN P;[
 350 NOP;PLA;LSR A;PHA;BCC OO5;LDA @CH"C";JSR #FFF4
 360:OO5 JSR #F7D1;];$P="02 @ ";P=P+LEN P;[;NOP
 362 PLA;LSR A;LDA @#01;ADC @#00;JSR #F80B
 364 JSR #F7D1;];$P=" MHz";P=P+LEN P;!P=#110A0A0D;P=P+4;[
 366 LDA #1FFC;JMP #FFF4
 370:VV0 BRK;BRK
 380:VV1;];?P=LL6%256;P?1=LL6/256;P=P+2
 390 NEXT A;PRINT $6
 400 LINK OO2;EXIT
```

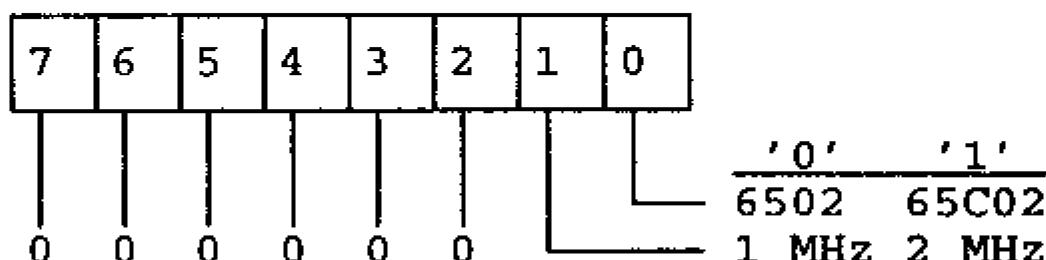
Eventueel saven met "TRUE-CPU.RUN"B00 C00 B94

Na het runnen van dit programma geeft OSBYTE#08 altijd de juiste processor informatie terug.

OSBYTE #08 opvragen processor gegevens

invoer : geen

uitvoer: accu bevat volgende informatie:



gewijzigde registers: Accu en vlaggen

Met vriendelijke groeten,

Roland Leurs
Hooge Hoeven 62
4191 MN Geldermalsen

Telefoon: 0345-572970
Fidonet: 2:285/226.9
Aconet: 77:8500/504.32

De Atom, een rekenwonder?

kanttekeningen bij een breuk in de Atom

door roland leurs

Eind vorig jaar verscheen een artikel van Leendert Bijnagte met daarin uitleg over de opslag van floating point getallen. Ofschoon het lijkt dat er alleen een typefoutje inzit, moet ik helaas constateren dat er enkele fouten in het verhaal zitten. Als u het hele verhaal gelezen en begrepen heeft dan moet ik u helaas teleurstellen: u heeft het niet begrepen...

Het begon zo onschuldig; halverwege pagina 15 van AN95-4 staat de binaire notatie voor -0.046875: 0111.1100.0100.0000.0000.etc. Echter de onderstreepte '0' blijkt in de praktijk een '1' te zijn. O, typefoutje ... Totdat ik voor het Atom terminalprogramma de floatingpoint uitbreidingen begon te schrijven. Bij de conversieroutines van Atom FP-notatie naar Intel FP-notatie en omgekeerd begonnen de problemen, sommige getallen werden wat de waarde betreft goed omgezet, maar het teken bleek verkeerd te zijn. Een lange zoektocht naar de ware gang van zaken begon. En met onderstaande resultaten.

In het nu volgende stuk geef ik alleen de verbeteringen aan. Op Henri's Unicorn BBS staat het hele artikel herschreven.

Blz. 13 Hoe dat op te lossen - het biasbit:
"Een '1' geeft aan dat er vermenigvuldigd moet worden, zijn antipode de '0' geeft aan dat delen de bedoeling is."

Exponent - Het biasbit is dus het teken van de exponent

Blz. 14 Voor de fractie
Het linkerbit is '0', wat aanduidt dat het getal positief is; ...

In het voorbeeld:
Nu nog even afwerken: het getal is positief, dus het tekenbit wordt '0'. De exponent hadden we bepaald op 7, dus het eerste getal is 1000.0111. Het bias bit staat op vermenigvuldigen, dus wordt '1'.

Blz. 15 De binaire notatie voor -0.046875 is 0111.1100.1100.00----0

Blz. 16 Voorbeeld: 0.75
"Het biasbit staat op delen .." Intel werkt niet met een biasbit maar noemt dit

gewoon het tekenbit van de exponent. Intel bedoelt met bias de basiswaarde die bij de exponent wordt opgeteld. Leendert noemt dat zelf halverwege deze bladzijde 'de basiswaarde'.

U ziet, het valt achteraf toch wel mee met de fouten, alleen zijn ze dermate belangrijk dat het hele begrip, de hele theorie er mee valt of staat. Eigenlijk ben ik wel blij met de fouten want nu heb ik alles precies moeten uitzoeken en nu begrijp ik het ook echt.

Nu u toch geïnteresseerd aan het lezen bent zal ik ook meteen de omrekenroutines die Leendert beloofd heeft eens tonen. Deze versie werkt op den Intel 80x86:

```
title   Conversie Intel8 -> Atom floating point

CSEG    SEGMENT PARA PUBLIC 'CODE'
        ASSUME CS:CSEG, DS:CSEG, ES:CSEG, SS:CSEG
        ORG 100H

START   PROC NEAR
        JMP  INTOM

ATOM    DQ  0
INTEL   DQ  -0.046875

INTOM:  MOV     AH,BYTE PTR INTEL[7]          ;LAADT MANTISSA IN AX:BX:DX
        MOV     AL,BYTE PTR INTEL[6]
        MOV     BH,BYTE PTR INTEL[5]
        MOV     BL,BYTE PTR INTEL[4]
        MOV     DH,BYTE PTR INTEL[3]
        MOV     DL,BYTE PTR INTEL[2]
        MOV     CX,11                         ;LAADT TELLER
LB1:    SHL     DX,1                          ;SCHUIF LINKS
        RCL     BX,1
        RCL     AX,1
        LOOP    LB1                           ;ELF MAAL DOEN
        SHL     AX,1                          ;EVEN OPSCHUIVEN
        MOV     CL,BYTE PTR INTEL[7]          ;LEES TEKEN VAN INTEL
        SHL     CL,1                          ;SCHUIF TEKEN IN CARRY
        RCR     AX,1                          ;SCHUIF TEKEN IN MANTISSA
        MOV     BYTE PTR ATOM[1],AH           ;ATOM MANTISSA WEGSCHRIJVEN
        MOV     BYTE PTR ATOM[2],AL
        MOV     BYTE PTR ATOM[3],BH
        MOV     BYTE PTR ATOM[4],BL
        MOV     AH,BYTE PTR INTEL[7]          ;LAADT INTEL EXPONENT
        MOV     AL,BYTE PTR INTEL[6]
        MOV     CL,4                          ;LAADT TELLER
        SHR     AX,CL                         ;BEPAAL EXPONENT
        AND     AX,7FFH                       ;ALLEEN ONDERSTE 11 BITS
        SUB     AX,1022                       ;ATOM EXPONENT BEREKENEN
        PUSHF                                 ;SAVE CARRY
        SHL     AL,1                          ;EVEN PLAATSMAKEN VOOR BIAS
        POPF                                  ;HAAL CARRY TERUG
        RCR     AL,1                          ;ZET CARRY ALS BIAS
        XOR     AL,80h                        ;INVERTEER
        MOV     BYTE PTR ATOM[0],AL           ;BERG ATOM EXPONENT OP
        RET
```

```
START   ENDP

CSEG    ENDS
        END     START
```

Het runnen van dit programma is ietsie onhandiger dan normaal is bij een pc. Start het programma met DEBUG INTOM.COM <enter> en vervolgens het commando G <enter> om het programma te starten. Vervolgens kunt u het resultaat onderzoeken met D 100 <enter>.
Voorbeeld van deze debug-sessie:

```
debug intom.com
-d 100
74BE:0100  EB 11 90 00 00 00 00 00-00 00 00 00 00 00 00 00   ...............
-g

Program terminated normally
-d 100
74BE:0100  EB 11 90 7C C0 00 00 00-00 00 00 00 00 00 00 00   ...|...........
-q
```

U ziet op de adressen vanaf 0103 de hexadecimale waarde staan voor Atom's notatie van het getal -0.046875. 7C C0 00 00 00 = 0111.1100.0000.0000.00---0; voor ons een bekend getal.
Het programma ATEL.COM werkt op dezelfde manier, alleen wordt daar een Atom getal naar Intel notatie omgerekend.

Aangezien het de afgelopen weken toch te koud is geweest om te schoffelen in mijne hof heb ik ook reeds een aantal floating point functies in de Atom terminal verwerkt. En dit werkt heel goed. Voorlopig alleen nog voor 80387 en hoger. Aan de uitbreidingen voor de 80287 wordt nog gesleuteld.

Een van de mooiste testprogramma's is deze:

```
10 REM GONIO GRAPHICS
20 ZERO;DIM M30,LL3,P(-1);PRINT $21
30[:LL0 \ Opgeven van floating point commando
40 LDA @#00;JSR #FFC8;LDA @#62;JSR #FFC8;TYA;JSR #FFC8;CPY @0;BNE LL1
50 JSR #FFC8;JSR #FFC5;RTS
60 :LL1 LDX @0
70 :LL2 LDA #605,X;JSR #FFC8;INX;CPX @5;BNE LL2
```

```
 80 :LL3 JSR #FFC5;STA #605,X;INX;CPX @10;BNE LL3
 90 RTS;];PRINT $6$12
100 Y=0;USR LL0
110 IF A=0 THEN PRINT "Geen coprocessor aanwezig !"';EXIT
120 IF A=1 THEN PRINT "80287 coprocessor aanwezig."';EXIT
130 IF A=2 THEN PRINT "80387+ coprocessor aanwezig."'
140 KADER 19,6,60,4,S,#3;CURSOR OFF
150 PRINT " Kies een grafiek of breek af met <ESC>"
160 $M=" Tangens  Sinus   Cosinus "
170 PDMENU 35,10,9,3,D,#47,$M,A
180 IF A=0 THEN TXMOD;EXIT
190 CLEAR 21;XSTYLE #F0F0;COLOUR 7;ESCAPE=OFF
200 XLINE 0,100,639,100;XLINE 0,250,639,250
210 TIME;PRINT '
220 FOR Z=100 TO 250 STEP 150
230 MOVE 0,Z
240 FOR X=0 TO 1279 STEP 2;%A=RAD X
250 IF Z=100 THEN GWINDOW 0,175,639,0;COLOUR 2
260 IF Z=250 THEN GWINDOW 0,349,639,174;COLOUR 3
270 IF Z=100 THEN Y=A+7;LINK LL0;GOTO 310
280 IF A=1 THEN %B=TAN %A
290 IF A=2 THEN %B=SIN %A
300 IF A=3 THEN %B=COS %A
310 DRAW (X/2),%(Z+%B*-75)
320 NEXT X
330 TIME;PRINT';GWOFF
340 NEXT Z
350 ESCAPE=ON;LINK #FFE3;TXMOD;GOTO 140
```

Dit programma laat afhankelijk van uw keuze een sinus, cosinus of tangens grafiek zien. Tevens krijgt u grofweg de tijd te zien die nodig is voor het tekenen van die grafieken. Met behulp van de coprocessor op een 80486dx2/66 pc kan de tijdwinst behoorlijk oplopen. In 20-50% van de tijd die de Atom zelf nodig heeft worden de goniometrische functies uitgevoerd en afgebeeld op het scherm. Verder kunnen we nu, indien die behoefte bestaat, gebruik maken van de Intel 64 bits floating point notatie, dus grotere nauwkeurigheid. Helaas niet toepasbaar in welk huidig Atom programma dan ook.

Er ligt nu nog een behoorlijke klus op ons te wachten. De floating point rom van de Atom moet nu aangepast of herschreven worden om gebruik te maken van de floating point commando's van de pc. Mocht u hier informatie over hebben of wilt u er aan meewerken laat het dan even weten.

Voorlopig kunnen we alleen op ietwat omslachtige wijze profiteren van de nauwkeurigheid en snelheid van pc's floating point processor...

U wilt ook experimenteren? De betreffende files die in dit artikel genoemd zijn kunt u halen op Henri's BBS; download het bestand "ATOMx87.ZIP". Dit bestand bevat de volgende files:

INTOM.ASM	source van conversieprogramma van intel8 -> atom fp
ATEL.ASM	source van conversieprogramma van atom -> intel8 fp
INTOM.COM	geassembleerde versie van INTOM
ATEL.COM	geassembleerde versie van ATEL
ATOM311.COM	terminalprogramma met 80387 floating point ondersteuning
*.RUN	diverse demonstratieprogramma's voor Atom met 80387
BREUK.TXT	gewijzigde versie van Leendert's artikel uit AN95-4

Met vriendelijke groeten,

Roland Leurs
Hooge Hoeven 62
4191 MN Geldermalsen

Telefoon: 0345-572970
Fidonet: 2:285/226.9
Aconet: 77:8500/504.32

Going round in circles

Soms zijn er dingen die als van zelf gebeuren. Soms doe je heel veel moeite, en dan laat het gewenste resultaat uiteindelijk toch nog te wensen over. Soms gaan dingen spontaan en ontstaat iets waar je als schrijver zelf ook verbaast over staat.
Zo was het eigenlijk nooit de bedoeling om een serie te schrijven over de achtergrond van punten, lijnen en dergelijke.
Maar ja, als daar nou vragen over worden gesteld, waarom dan toch maar weer niet in de printer klimmen en wat leesvoer voorbereiden.
Vandaar hier het derde, en wie weet niet het laatste, artikel over de achtergrond van figuren. Dit keer de cirkel, u weet wel van die figuren die gemaakt worden met die dingen die wijdbeens door het leven gaan en graag in het middelpunt staan, ofwel een passer. Hier wordt u een analyse gepresenteerd van het atomaire equivalent van de passer.

Het principe

Er zijn diverse technieken beschikbaar voor het tekenen van cirkels, als we die van de genoemde passer buiten beschouwing laten.
In tegenstelling tot lijnen, die een constante hoek hebben, hebben cirkels lijntjes die constant van hoek veranderen. Die hoek ligt dan tussen 0 en een oneindig groot getal. In het geval van een constante hoek, heeft de programmeur het voor het zeggen welk punt gevuld wordt. In een van de vorige afleveringen heeft hierover meer gestaan, ik bedoel namelijk de bepaling van het 'Errorlevel'. Bij cirkels is een dergelijke vrijheid niet meer mogelijk, anders gaat het meer op aardappels lijken dan het keurige ronde zoals dat van een pizza, of om wat dichter bij de Atom te blijven: een CD-Rom.

Wat is een cirkel

Een cirkel bestaat uit een middelpunt en een afstand van het middelpunt naar de omtrek. Deze laatste worden gemakshalve even als X en Y beschouwd.
En omdat ik het (voor velen helaas) niet kan laten er toch nog een beetje middelbare school wiskunde tegenaan te gooien, wordt een cirkel voorgesteld door de vergelijking:

Radius^2 = X^2 + Y^2

Natuurlijk overbodig te zeggen dat in deze vergelijking geldt dat voor iedere X twee waarden van Y horen, namelijk het positieve deel en ook het negatieve deel. Want 2x2 is gelijk aan -2 x -2. Ja, toch?
Wie dat nog even in officiële notatie wil zien:

Y= +/- (Radius^2 - X^2)^1/2

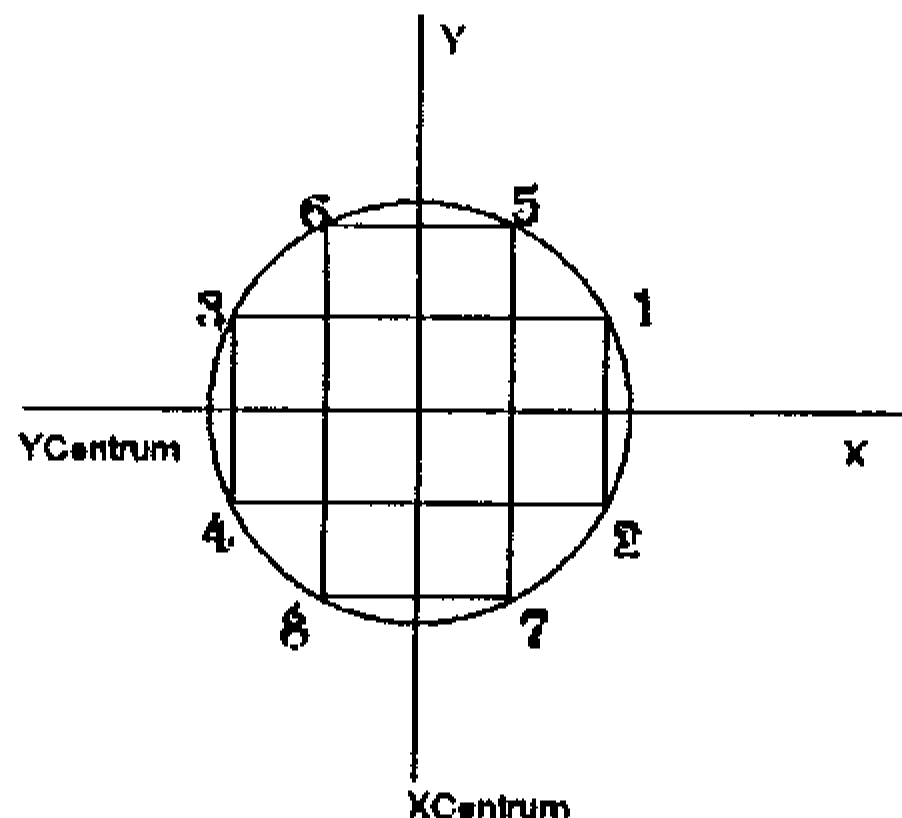
Wie wil kan in Basic even een programma in elkaar knutselen en voor alle waarden de straal van de cirkel berekenen. Ik ben benieuwd of we dat in een van de volgende nummers van AN mogen verwelkomen.

Het gebruik van symmetrie

Een cirkel is een symmetrisch geheel. Snij hem in tweeën, en spiegel het op de snij as en voila, daar is weer een cirkel.
Vanwege deze symmetrie, volstaat het om slechts 180 graden van een cirkel te berekenen en de rest op de X as te spiegelen. Dat scheelt al heel wat rekenwerk.
Maar er is meer. Ook bij een fikse snee in de Y as, is het nog steeds mogelijk om er weer een cirkel van te maken. Dus 0 tot 90 graden zou ook goed moeten gaan: eerst een kwart berekenen, dan dit kwart spiegelen en vervolgens de ontstane helft spiegelen.
Verder halveren is ook nog mogelijk: met een overspanning van 0-45 graden, is het mogelijk om een hele cirkel bijeen te spiegelen. Net als de partjes van een niet al te bedenkelijk merk sinasappel, passen al deze partjes bijeen en vormen samen een cirkel.
In bijgevoegde afbeelding is dit weergegeven, niet de sinaasappel, maar het principe van de acht segmentjes.

In onderstaande tabel wordt weergegeven welke vergelijkingen er nodig zijn, om tot bovenstaande cirkel te komen. Deze vergelijkingen worden doorwandeld in een programma van 0 tot 45 graden.

X1 = X_centrum + X	Y1 = Y_centrum + Y	Origine
X2 = X_centrum + X	Y2 = Y_centrum - Y	X spiegel
X3 = X_centrum - X	Y3 = Y_centrum + Y	Y spiegel
X4 = X_centrum - X	Y4 = Y_centrum - Y	X/Y spiegel




```
X5 = X_centrum + Y   Y5 = Y_centrum + X        45 spiegel
X6 = X_centrum - Y   Y6 = Y_centrum + X        45,X spiegel
X7 = X_centrum + Y   Y7 = Y_centrum - X        45,Y spiegel
X8 = X_centrum - Y   Y8 = Y_centrum - X        45,X,Y spiegel
```

Het tekenen van cirkels met behulp van een cirkel algoritme

Het tekenen van een cirkel gebruikmakend van de genoemde vergelijking, lijkt op dat van het trekken van lijnen: een coördinaat wordt gebruikt als een onafhankelijke variabele en de oplossing wordt berekend op basis van de variabelen.
Als parameter bij een cirkel wordt meestal het middelpunt opgegeven en die radius. Voor het oplossen in de genoemde vergelijking, wordt het middelpunt gebruikt en de straal. In dit voorbeeld gaat dat op als de coördinaten tevens op het gewenste middelpunt van de cirkel liggen. Dit is meestal niet het geval, want cirkels moeten her en der op het scherm komen. Vandaar dat er een offset moet worden opgeteld. In onderstaande formule is dat verder uitgewerkt:

```
X_cirkel = X + X_centrum
Y_cirkel = Y + Y_centrum
```

De cirkel wordt getekend door optimaal gebruik te maken van de symmetrie van het figuur. We gebruiken hier de X as als onafhankelijke variabele. In het volgende algoritme varieert de X projectie van 0 tot 45 graden. De X projecties zijn afgeleid van de eerder gegeven formule.

Bepalen van X limieten

Met behulp van een voorbeeld, worden hier de respectievelijke onder en bovengrens van de waarden bepaald.

XStart = Radius x cos(theta), hier is theta gelijk aan 0 graden. In waarden uitgedrukt betekent dit:
Xstart = Radius x cos(0) = Radius, want cos(0 graden) is 1

XEinde = Radius x cos(theta), hier is theta gelijk aan 45 graden, 1/8 deel van een cirkel, die altijd 360 graden is.
XEinde = Radius x cos(45) = Radius x 0.7071

Rondje van de zaak

Om een cirkel te krijgen, zijn de volgende stappen nodig:

1. Bepaal de X_start en X_einde met de gegeven vergelijking
2. Los de vergelijking op met een positieve waarde voor Y, met X als vast gegeven.(U weet nog wel: X = radius)
3. Vertaal de X index en de gevonden oplossing om vervolgens voor Y om de achtvoudige symmetrie te benutten
4. Zet de acht punten voor de gegevens van punt 3.
5. Verhoog de X variabele.
6. Als de waarde van X niet groter is dan (Radius x 0.7071), ga dan verder met stap 2.

Hoewel deze methode verdacht veel op ganzebord lijkt, heeft het toch een aantal minder prettige eigenschappen. Voor die hoeken waar het resultaat groter dan 1 is, dus waar de hoek oneindig groot dreigt te worden, is ons cirkeltje niet meer rond, maar wat ellips achtig. En dat was nou net niet de bedoeling.

Een ander belangrijk nadeel is dat voor iedere oplossing twee vermenigvuldigingen, een optelling, en wat nog veel erger is, en een worteltrek berekening nodig is. En de vermenigvuldiging betreft een floating point getal en dat is, gezien de andere stukken in AN, een reeds bekend drama in performance land.
Om over het wortelen maar te zwijgen want, hoewel daar een aantal goede technieken voor beschikbaar zijn, beperken deze methoden zich van langzaam tot zeer langzaam.
Maar er is hoop.

Het tekenen van een cirkel met behulp van parametrische vergelijkingen.

Een tweede techniek is het gebruikmaken van parametrische vergelijkingen. Deze klinken nogal ingewikkeld, maar staan eenvoudigweg weergegeven in onderstaande vergelijking.

X = Radius x Cos(theta)
Y = Radius x Sin(theta)

Theta is in dit geval de hoek van 0 tot 360 graden. In deze vergelijking is theta de variabele. Het tekenen van een cirkel wordt dan gedaan door theta met een constante waarde te verhogen.
De punten die op de locatie X en Y worden getekend, vormen tezamen een prachtige cirkel. Des te kleiner de stapgrootte, des te nauwkeuriger de cirkel. Deze waarde kan berekend worden met het gegeven dat de omtrek van een cirkel vastgelegd is met de berekening dat 2xPixRadius is. Is de radius van een cirkel 100, dan is de omtrek van die cirkel: 2 x 3.14 x 100 = 628.
Om, in dat geval, een cirkel op het scherm te krijgen, waarbij alle puntjes aaneengesloten dienen te zijn, is het noodzakelijk om 628 punten te berekenen. Hierbij wordt aangenomen, dat er gewerkt wordt met een stapgrootte van 1 hele graad. Zeker voor grotere cirkels bij een hogere resolutie is dat niet voldoende. Hierbij wordt de stapgrootte dus kleiner en de rekentijd langer.

Als gewerkt wordt met parametrische vergelijkingen, is het van essentieel belang dat het middelpunt van de cirkel als origine wordt beschouwd. Is het toch elders op het scherm, dan moet bij het **resultaat** de eventueel benodigde offset worden bijgeteld.

Een lekkere versneller voor dit soort grappen is het toepassen van een SIN en COSIN tabel. Frans van Hoessel heeft in een grijs verleden hierover een fabelhaft stukje geschreven. Hier werd het een tabel gebruikt voor het opzoeken van de sinus en het complement genomen voor het bepalen van de cosinus. Als klap op de vuurpijl werd nog gebruik gemaakt van een superhandige vermenigvuldigingstabel. Wie weet wijd ik daar nog eens een apart artikel aan. Watch your mailbox!

Het tekenen van een cirkel met deze methode gebeurt op basis

van de volgende stappen:

1. Zet de waarde van theta op 0
2. Los de vergelijking voor X en Y op gegeven de waarde van theta.
3. Vertaal de oplossing en gebruik de voordelen van de achtvoudige symmetrie.
4. Teken de acht puntjes, zoals hierboven berekend.
5. Verhoog de waarde van theta, zodanig dat er een aaneengesloten belijning ontstaat aan de rand van de cirkel.
6. Herhaal dit circus totdat theta groter is dan 45 (graden)

Volkomen overbodig te zeggen, dat ook deze methode niet de frisheid van snelheid uit zal ademen. Het berekenen van een sinus in assembler is op zich al een uitdaging, laat staan om daar ook nog iets van een cirkel van te maken. Maar er is nog meer hoop.

Bresenham's cirkel

Een derde techniek is ontdekt door Bresenham en is vergelijkbaar met het door hem ontwikkelde lijn algoritme, wat u ook nog kent uit een van de vorige afleveringen.
Net als het lijn-algoritme, is ook deze techniek gebaseerd op een beslissingsvariabele, zeg maar de errorlevel. Het besluit (verhogen, verlagen) is gebaseerd op een beslissingsvariabele D, waarbij D wordt geïnitialiseerd op basis van de vergelijking:

D = 3 - (2 x Radius)

Zowel X als Y kunnen hier dus worden beschouwd als onafhankelijke waarde, vanwege de symmetrie van een cirkel. Stel dat X een onafhankelijke waarde heeft. X wordt gelijk gesteld aan 0 en wordt gedurende het iteratieve proces verhoogd. X varieert van 0 tot een waarde die kleiner is dan Y. Y op haar beurt, wordt geïnitialiseerd op de waarde van de radius. Elke iteratie door het algoritme tekent uiteindelijk 1 punt, pixel. Het teken van D bepaalt dan uiteindelijk of Y moet worden verlaagd. Het schrijft eveneens voor welke aanpassing er zo nodig aan D moet worden gedaan. Dit wordt in onderstaande vergelijking uitgelegd.

D = D + (4 x X) + 6   als D<0

D = D + 4 x (X - Y) + 10 als D>=0

Vanwege het gegeven dat de X variabele varieert van 0 tot Y, wordt slechts een achtste gedeelte van de cirkel berekend. De symmetrie zorgt voor de rest van de sinaasappel, herstel: cirkel.
Even een toelichting waarom dit zo zou zijn. X varieert van 0 tot Y. Uit de sinustabel kennen we het gegeven dat als de X gelijk is aan Y, dit alleen waar is voor een hoek van 45 graden. En daar waren we nou net naar op zoek. Slimme jongen, die Bresenham.

Hoewel de echte criticus nu zal zeggen dat ook dit niet super-snel zal zijn, kan ik haar geruststellen door te zeggen dat alle vermenigvuldigingen gaan over machten van 2 en die zijn weer met een eenvoudige shift te realiseren. Hele slimme jongen, die Bresenham.

En nu verder

Wellicht in een volgende Atom een voorbeeld van het Bresenham algoritme voor cirkels. Het is de moeite waard om dit eens nader uit te werken.

De geschiedenis van het floating point project

In een van de vorige nummers heb ik in een vlaag van verstandsverbijstering allerlei zaken lopen opnoemen die nog mogelijk zijn met de Atom in PC.
Een daarvan was het uitbouwen van de floating point activiteiten naar een eventueel geïnstalleerde mathematische co-processor.
Dat artikel is weer met het Atomclub bekende enthousiasme ontvangen. Elders in dit blad kunt u de ervaringen en de bijbehorende inspanning van Roland op dit vlak verder lezen, maar ik heb even in de keuken van de meester mogen gluren en draai nu ook met de Atom terminal die ook de x87 ondersteund. Dat ik daar natuurlijk ter plekke lirisch van ben geworden, dan behoeft nauwelijks verder toelichting. Anders had dit artikel natuurlijk nooit het levenslicht kunnen aanschouwen.
In dit artikel geen verder toelichting over de achtergrond en de implementatie, want dat doet Roland al in zijn artikel. Maar gewoon even een verslagje van een avondje klungelen met het wiskundeboek van vroeger op de schoot. En wat voor voordelen deze softwareuitbreiding nu werkelijk biedt.

Het voordeel

Het grote voordeel van het rekenwerk uitbesteden aan de x87, is dat van de snelheidswinst.
In zijn dagen behoorde de Atom tot een der snelsten, maar net als tijd, is ook snelheid een betrekkelijk begrip geworden.
Wie in de jaren '80, met name het begin ervan, had verteld dat er ooit een processor op 133 MHz zou draaien, werd acuut verdacht een doorgeschoten FM amateur te zijn, die niet gehinderd door enige kennis van zaken was beland in het digitale tijdperk.
Het leuke van de x87 is dat deze vanaf de 486DX reeds ingebakken zit in de chip en nu zonder financiële inspanning voor een Atom-in-PC uitbreiding kan worden aangewend. Enfin, dat weet u natuurlijk al, want dat heb ik al een keer geschreven.

Hoe te handelen vanuit de Atom

Roland heeft reeds fraaie conversieroutines gemaakt om de floating-point notatie van Atom naar PC en vice versa te doen. Ook zijn de functies reeds in de Atom terminal, dat wil zeg-

gen: de dos versie, geïmplementeerd.
Brengt me tevens op het feit dat in de vorige AN enige bits in mijn floatingpoint verhaal zijn omgevallen. Laten we het er maar op houden dat mijn printer de zaak onduidelijk heeft gemaakt, want om toe te geven dat ik zelf wat fouten heb gemaakt, valt me zwaar.

Het voorbeeld

Ik heb in een oud boekje een leuk programma van een dito leuk plaatje opgezocht. Het enige nadeel van leuke plaatjes is dat ze in de regel ook veel rekentijd kosten. Dus waarom niet even snel krasse, krasse om de x87 wat werk te laten doen. Zo gezegd, zo gedaan. En omdat ik een lui mens ben, heb ik alleen de CPU knagers zoals de SIN en dat soort vrienden geëlimineerd.
Toch is het resultaat verbluffend: ik, dat wil zeggen de AtomTerminal van Roland, scoorde een winst van ruim 100%. Niks geen geknabbel met een paar procenten door wat EDO ram, nee gewoon twee keer zo snel. En dat terwijl het nog maar voor een deel van de x87 gebruik maakt. Pas op voor wat nog gaat komen!

Listing

Hier dan het gebrabbel. De originele regel is steeds vervangen door de assembler antipode en omwille van wat leesbaarheid, heb ik in een rem de originele instructie laten staan.

Zo hier en daar is een kleinigheid aangepast, maar dat heeft met name te maken met het feit dan Meneer Van Dale Wacht Op Antwoord, u weet (nog) wel de regel die aangeeft dan machtsverheffen, vermenigvuldigen, delen, optellen en aftrekken ook daadwerkelijk in die volgorde moeten geschieden. Dit wordt zonodig overruled door er haakjes omheen te plaatsen en met name dat wordt nog door de interpreter uit de rom verwerkt.

Het uiteindelijke resultaat van wat binnen de haken staat, wordt doorgegeven aan de FP rom. Vandaar dat ik zo hier en daar even moest splitsen om eerst wat tussen de haken staat te berekenen en daarna dit argument door te geven aan de Terminal.

Tijd is betrekkelijk

Heb wel geduld als u dit programma draait, want ook al gaat het twee keer zo snel, het kan nog steeds (m)enige tijd duren voordat het prachtige beeld volledig is gevuld.

```
 10 REM ROTATING DEMO
 20 ESCAPE=OFF
 30 %A=2.5;%B=1.3
 40 C=#00;GOSUB c;REM TEST AANWEZIGHEID COPROCESSOR
 50 A=#00;LINK #FFC8
 60 USR #FFC5;PRINT A';IF A=0 THEN PRINT "Geen 80x87
    aanwezig !"';GOTO 980
 70 IF A=1 THEN PRINT "8087/80287 aanwezig"'
 80 IF A=2 THEN PRINT "80387+ aanwezig"'
100 P=#7000;DIM LL2;[:LL0
    \ BEREKEN %C={funtie} %A:
110 LDA @#00;JSR #FFC8;LDA @#62;JSR #FFC8;TYA;JSR #FFC8
    \ loopje voor het doorgeven van %A aan de PC:
120 LDX @#00;:LL1 LDA #605,X;JSR #FFC8;INX;CPX @5;BNE LL1
    \ loopje voor het doorgeven van het resultaat naar %C:
130 LDX @#00;:LL2 JSR #FFC5;STA #60F,X;INX;CPX @5;BNE LL2
140 RTS;]
200 CLEAR 22
205 TIME;P.'
300 A=320;B=A*A;C=240
310 FOR X=0 TO A
320    REM S=X*X; %P=SQR(B-S); %I=-%P
325    S=X*X; %A=(B-S);Y=7;LINK LL0; %P=%C;%I=-%P
330    REM CALCULATE R = DISTANCE FROM CENTRE. Q=HEIGHT
335    DO
340      REM %R=SQR(S+%I*%I)/A
345      %A=S+%I*%I;Y=7;LINK LL0;%R=%C/A
350      REM %Q=(%R-1)*SIN(24*%R)
355      %A=24*%R;Y=9;LINK LL0;%Q=(%R-1)*%C
360      %Y=%I/3+%Q*C
370      FIF %I=-%P %M=%Y; GOTOd
380      FIF %Y>%M %M=%Y; GOTOe
390      FIF %Y>=%N GOTOf
400d     %N=%Y
410e     %Y=C+%Y
420      PLOT 13,(A-X), %Y;PLOT 13, (A+X), %Y
430f     %I=%I+4; FUNTIL %I>=%P
440 NEXT
```

```
450 TIME;P.'
980 ESCAPE=ON;EXIT
990cA=#00;LINK #FFC8;A=#62;LINK #FFC8;A=C;LINK #FFC8;RETURN
```

Tenslotte

De floating-point in deze mode wordt nu nog aangeroepen via een LINK call, met de juiste parameters in de floating point variabele %A. Het resultaat wordt teruggeschreven in de variabele %C. Het soort instructie wat moet worden gedaan (sin, cos, tan etc.) wordt nu nog even als parameter meegegeven: een 9 voor de functie sinus en 7 voor de functie SQR, het worteltrekken. Dit wordt in de Link LLO allemaal geregeld.
Let u er ook nog even op dat de FP variabelen niet, zoals dat bij de Atom gewoon was, op #2800 en verder staan maar op #0600 en verder. Reeds gedocumenteerd in de respectievelijk verschenen handleidingen, maar bij niet dagelijks gebruik van de FP variabelen net iets om te vergeten.

In een later stadium, en volgens Roland schijn ik dat te moeten maken, wordt de Floating point rom nog aangepast zodat u gewoon weer SIN in kan toetsen en de rom babbelt dan onder water via de terminal met de x87. En dat zonder gebruik te maken van de hulpvariabelen %A en %C.

Fijn dat de Atom In PC er is. Wordt mijn x87 toch nog gebruikt anders dan voor mijn spreadsheet!
Wie (nog) niet in het bezit is van een x87, wordt verwezen naar het emulatieprogramma wat beschikbaar is op het BBS van Henri. Ook met het emulatieprogramma zijn toch nog zeer verbluffende resultaten haalbaar.

Leendert

Contributie 1996 nog een keer? Ja, alweer!

Zo begon in het vorige nummer een artikel waarin u verzocht wordt om de contributie te betalen als u lid wil blijven. Ik zie aan het gebruik van de leestekens in dat artikeltje dat het geschreven is door Bruno. Ik ken inmiddels ook zijn bezorgdheid omtrent de AtomClub. Hij vroeg het zich in december 1993 ook al af. Ergens heeft hij het gevoel dat de AtomClub een aflopende zaak is.

Gelukkig denk ik er heel anders over. De Atom blijft iets hebben dat geen enkele andere computer mij kan bieden: duidelijkheid. Al jaren werk ik (ook beroepsmatig) met PC's en de meest wilde operating systemen. Maar al dat moois geeft mij geen compleet genoegen; als er een lijn op het scherm getekend wordt dan wil ik weten hoe dat werkt. Het lijkt zo gewoon om een lijn te tekenen maar welke routines zijn daar voor nodig? Eenzelfde voorbeeld is de floating point. Start een calculatorprogramma en toets maar in 2.3 + 1.1 en het antwoord verschijnt op het schermpje. Maar hoe zit dat?
Deze antwoorden vind ik niet terug bij een HCC vereniging; aangezien men daar vast al bezig is om een pre-beta-versie van Windows'97 te testen. Of een Laserjet 6 die 16 miljoen kleuren in 3200x3200 dpi voor slechts 5000.- kan afdrukken. Geen enkele hobbyist die daar op zit te wachten.

Neen, vrienden van de Atom, die antwoorden vind ik hier in dit boekje dat voor u ligt. Hier wordt uitgelegd hoe een kleurenkaart verbeterd kan worden. (Heeft u zelf ooit een draadbrug gelegd van pin 2 naar pin 97 op uw VGA controller?) In dit blad wordt ook beschreven hoe u uw eprompprogrammer kunt bouwen. En hoe lijnen getekend worden en de opslag van floatingpoint getallen is voor u ook geen geheim meer.

Echter daar schuilt het gevaar waar Bruno bang voor is. Het niveau stijgt en stijgt. Als niemand er iets van begrijpt is het voor de AtomClub ook niet interessant meer. Leden haken af: "Het is mij te ingewikkeld" is hun (terechte) excuus. En zie daar voor 1997 komen de laatste vier contibutie-gelden bij Thijs binnen....

Laten wij met zijn allen voor 1996 nogmaals zien dat we ook eenvoudiger bezig kunnen zijn; laat zien waar U mee bezig bent. Want er wordt nog volop gesoldeerd en geprogrammeerd; dat is op de Landdagen wel te zien. Steeds weer zie ik mensen die iets nieuws hebben en steeds weer komt die ene meneer naar mij toe met een lijstje met vragen. Dat is toch geweldig! Maar wacht niet tot de landdag. Gebruik Atom Nieuws eens om wat te vragen of te tonen.

Enkele leden hebben via het Registratieformulier hun wensen kenbaar gemaakt door aan te geven welke hardware en software uitbreidingen voor hun interessant zijn. En volgens mij zijn zij niet de enige met die specifieke behoefte. Laten we eens even zien wat er zoal gewenst wordt in Atom-liefhebbend-Nederland:

Midi aan de Atom(-in-PC)
Cassette interface voor Atom-in-PC

Chemische en natuurkundige simulaties
Geheugenuitbreidingen
Z80-kaart aan Atom-in-PC
GDOS aan Atom-in-PC
Een extreem project: 6502 kaart voor Archimedes/RiscPC

Maar er is meer om over na te denken volgens deze Aconet bijdrage van Henri Derksen:

Ik heb al jaren gepropageert dat de Big Ben Club en de Acorn Atom Club best wat meer zouden mogen gaan samenwerken. Zo heb ik al meer dan eens voorgesteld om als intermediair tussen beide clubs op te treden. Ik ben immers in beiden actief.
Ik hoop dat die samenwerking nu ook wat meer zal gaan gebeuren. Anders blijft er straks nog maar een alternatief over, namelijk de Acorn GG binnen de HCC of doodbloeden/opheffen van zowel de Acorn Atom Club als de Big Ben Club, en dat zou zonde zijn.

Ik hoop dat de 8-bits liefhebbers binnen de Big Ben Club (voornamelijk BBC + Electron) en de leden van Acorn Atom Club elkaar zullen weten te vinden. Zo is er al tijden gevraagd om een 8-bits werkgroep binnen de Big Ben Club. Er is wel belangstelling voor (de resultaten van) die werkgroep maar bijna niemand wil daarin plaatst nemen. Ik wil op een zijpaadje daaraan wel meewerken, maar is een zware hoofdtaak daarin niet voor mij weggelegd.

Moge AcoNet ertoe bijdragen dat de samenwerking BigBenClub <-> Acorn Atom Club gaat lopen en eventueel ook nog die 8-bits-werkgroep van de grond komt. Zo zouden (voormalige) Atom Club BestuursLeden misschien wel het bestuur van die BigBen 8-bits werkgroep kunnen/willen gaan vormen ?????

Misschien kan het HoofdBestuur van de Big Ben Club er ook wat aan (gaan) doen dat een betere integratie mogelijk wordt. Daar varen we allemaal wel bij denk ik.
Van mijn kant wil ik een bemiddelaarsrol spelen omdat ik beide kampen begrijp. En laten we ons dan gezamelijk warmen aan een wat groter kampvuur dan aan twee kleinere. Met wat meer geduld en luisteren naar elkaars argumenten en de ander ook de ruimte geven/laten die hij/zij nodig heeft, moet er toch een samenwerking op poten te zetten zijn. Zonder dat de ene club de andere verkettert omdat het bestaan van de ander en diens systemen niet erkend (zou) worden. Grote onzin natuurlijk, die systemen zijn er nu eenmaal, en zelfs bloedverwant (zelfde ouders ;-).

We kunnen een hoop van elkaar leren als we elkaar de ruimte geven. Zo noem ik al jaren bewust alle machines in mijn uitlatingen zoals op UniCorn BBS. Het is daar een ware smeltkroes van om het even welk acorn systeem dan ook.

Daarbij denk ik wel dat de Acorn Atom Club zo realistisch zal moeten zijn om te beseffen dat er NU geen/weinig mensen zullen zijn die nu nog met een Atom zouden willen beginnen. Daarentegen kan niet hard genoeg worden gewerkt aan het behouden van het oude Atom Enthouiasme en ik vind dat de BigBenners die ook zullen moeten gunnen

en weten te waarderen. Bijna iedereen is op een (oude) Acorn begonnen. En lang niet iedereen wil meedoen aan die Hype van Windows en de Windhoos laat staan InterNet. Je moet degeen die aan oude simpele systemen vasthouden en daarmee kunnen doen wat ze willen doen ook in hun waarde laten. Alleen dan krijg je een gezellige club waarin iedereen zich thuis voelt. Mijn strijdkreet is niet voor niets: Soms samen, soms ieder apart.

Ik hoop weer voldoende stof voor dicussie te hebben opgebracht. Als in beide clubs deze discussie gelijktijdeig en netjes wordt gevoerd is er misschien hoop voor een veel nauwere samenwerking.

Aldus Henri.

Het stukje van Henri kwam toevallig ter sprake naar aanleiding van een discussie in de BigBenArea van Aconet. Ik reageerde daar op en prompt kwam deze boodschap retour. Inmiddels hebben we daar al wat verder op gemaild (een antwoord is ook in de AtomMessageArea geplaatst) en er blijken meerdere mensen van de Big Ben Club interesse te hebben in samenwerking. Het zijn niet alleen Henri en ik die dat willen!

Vooral Henri's opmerking dat wij dan niet moeten verwachten dat de halve Big Ben Club een Atom gaat bouwen. Maar we kunnen wel parallel werken aan hardware projecten. Dit gaat met name op voor de 8 bits Acorns die niet Atom heten. Een via kan op bijna dezelfde wijze aan een BBC gekoppeld worden als aan een Atom. En ook de software (zeker voor assembler) kan vrij goed vertaald worden van de ene naar de andere machine. Plus dat vele Atomisten ook een Electron in de kast hebben liggen die ze ooit voor een prikkie hebben gekocht.

Henri gaat op één punt nogal ver, namelijk dat de Big Ben Club de Atomclub overneemt; zij het in een aparte groep: de 8-bits werkgroep. In dat geval moeten we goed bedenken wat we willen. Als we inderdaad samengaan moeten we o.a. rekening houden met overname van Atomclub eigen vermogen en goederen, contributie aanpassing en het mogelijk opheffen van Atom Nieuws.

In ieder geval wil ik hierbij het bestuur van de federatie verzoeken om dit punt mee te nemen in de volgende jaarvergadering en natuurlijk het verzoek van de leden om ook hierop te reageren.

Ik hoop zelf dat we het bestaan van de Atomclub naar ieders tevredenheid kunnen voorzetten en daar extra een goede samenwerking met de 8-bits werkgroep van de Big Ben Club aan kunnen toevoegen.

Met vriendelijke groeten,

Roland Leurs
Hooge Hoeven 62
4191 MN Geldermalsen

Telefoon: 0345-572970
Fidonet: 2:285/226.9
Aconet: 77:8500/504.32

Atom in PC reset extension vector

door roland leurs

Een van de weinige "undocumented features" die de Atom in PC u te bieden heeft is de mogelijkheid om de reset routine uit te breiden. In de handleiding en in wat artikelen is hier vaag wat aandacht aan besteed. In dit artikel wordt het voor eens en voor altijd duidelijk beschreven. Tevens krijgt u de voorgestelde aanpak te zien hoe routines deze vector mogen gebruiken.

Reset ... en dan ?

De standaard Atom resetroutine (vanaf #FF3F) test op adres #1FFD of daar de waarde #F0 staat. Zo ja, volgt een JMP (#1FFE). In woorden: er wordt gesprongen naar het adres dat in #1FFE/#1FFF is opgeslagen. Voor EPOS/3 is dat #1FB0. Daar staat een aanvullende resetroutine die het o.a. mogelijk maakt om de operating systeem software in RAM te aktiveren, de tekstkleur wordt aangepast (kleurcode is opgeslagen in adres #1FFC) en ook de mogelijkheid om de resetroutine naar eigen inzicht uit te breiden.

In de aanvullende resetroutine worden twee soortgelijke testen uitgevoerd voor een uitbreiding van deze aanvullende routine. Als op adres #4FA een machinetaal instruktie 'JMP xxxx' staat (waarbij xxxx het adres is waar eigen uitbreidingen staan) volgt een 'JSR #4FA'. Die eigen uitbreiding hoort natuurlijk te eindigen met een 'RTS' instruktie. De Atom keert terug naar de aanvullende resetroutine in het #1Fxx blok. Vervolgens wordt bovenstaand ritueel nogmaals uitgevoerd voor adres #4F7.

Eigen uitbreidingen.

We hebben dus drie mogelijkheden om in te grijpen in de resetroutine. Hiervoor gelden de volgende gedragsregels:

Type	Controlebyte	Primair gebruik
OS_RES_EXT	#1FFD = #F0	OS reset uitbreiding
SYS_RES_EXT	#4FA: JMP ssss	Systeem reset uitbreiding
USR_RES_EXT	#4F7: JMP uuuu	User program reset uitbreiding

OS_RES_EXT — Het is niet toegestaan om deze sprong te wijzigen, tenzij 100% compatibileit met het originele systeem is gegarandeerd. In praktijk: afblijven!

SYS_RES_EXT — Deze routine biedt de mogelijkheid om automatisch operating systeem uitbreidingen te aktiveren. Bijvoorbeeld patches en uitbreidingen van systeem calls. Voorbeelden hiervan zijn XMSDRV.RUN (toevoegen XMS osbyte-calls) en TRUE-CPU.RUN (patch voor OSBYTE call #08).

USR_RES_EXT — Deze vector is geheel vrij voor eigen programma's. Een van de meest zinnige toepassingen is het automatisch starten van een (menu-) programma nadat de Atom gereset is.

Implementatie.

Het programma weet nooit of er reeds andere routines gebruik maken van de uitbreidings-mogelijkheden. In het voorbeeld voor SYS_RES_EXT zien we al dat er mogelijk twee routines gebruik maken van dezelfde vector. Dat houdt in dat de programmeur de vector bij installatie moet controleren en, indien reeds in gebruik, opslaan. Nadat tijdens de systeemreset het initialisatieprogramma doorlopen is zal er aan de hand van de opgeslagen informatie gecontroleerd moeten worden of er nog een andere routine aangeroepen moet worden.

In een programmavoorbeeld kan dat er zo uitzien:

```
   \ PROGRAMMA INSTALLATIE
290:OO2 LDA #4FA;CMP @#4C;BNE OO3
300 LDA #4FB;STA VV1;LDA #4FC;STA VV1+1
310:OO3 LDA @OO0%256;STA #4FB;LDA @OO0/256;STA #4FC
320 LDA @#4C;STA #4FA
```

```
380:VV1;];?P=LL6%256;P?1=LL6/256;P=P+2
```

In regel 290 wordt gecontroleerd of adres #4FA de waarde #4C bevat; #4C is de opcode van de JMP instruktie. Indien geen JMP aanwezig wordt "regel 300" overgeslagen. Vanaf regel 310 wordt de nieuwe SYS_RES_EXT gezet. Als in adres #4FA wel #4C staat wordt de inhoud van de SYS_RES_EXT gekopieerd naar adres VV1. In regel 380 wordt ervoor gezorgd dat bij het starten van het programma op adres VV1 een verwijzing staat naar een RTS instruktie.
De routine vanaf label OO2 wordt uitgevoerd tijdens de installatie van de reset uitbreiding.

Onderstaande routine wordt uitgevoerd na een reset:

```
   \ INITIALISATIE
224:LL6 RTS
230:OO4 CMP @#08;BEQ LL0;JMP (VV0)
260:OO0 LDA #220;STA VV0;LDA #221;STA VV0+1
270 LDA @OO4%256;STA #220;LDA @OO4/256;STA #221
280 JMP (VV1)
```

In het installatiegedeelte hebben we gezien dat de SYS_RES_EXT wijst naar de routine OO0. De inhoud van VV1 wijst ofwel naar een RTS instruktie ofwel naar een andere uitbreiding van deze vector.
Bij het binnenkomen van de uitbreiding wordt eerst uitgevoerd wat er moet gebeuren (in dit voorbeeld is dat het omzetten van de osbyte-vector naar een eigen routine). Als dat af is wordt gesprongen naar het adres dat opgeslagen is in adres VV1. Omdat tijdens de installatie ervoor gezorgd is dat daar de juiste waarde staat hoeven we ons verder niet druk te maken of er andere reset uitbreidingen aanwezig zijn, als dat zo is wordt er automatisch heen gesprongen.

USR_RES_EXT voorbeeld

Een voorbeeld van USR_RES_EXT toepassing is het starten van de !BOOT file die in de directory \ATOM moet staan. Dit korte programma maakt een nog korter programma aan op adres #7F00. Achter dat programmatje komen de machinetaal instrukties te staan om de textpointer ?18 op #7F te zetten en het programma te starten.

```
10 REM !BOOT-BOOTER
20
30 P=#7F00
40 !P=#0A000D;$(P+3)="MENU";P=P+3+LEN(P+3)
50 !P=#FF0D;P=P+2
60 C=P;[ LDA @C/256;STA #12;JMP #CE86;]
65 P=#4F7;[ JMP C;]
70 PRINT ''"!BOOT-BOOTER V1.00 is geinstalleerd."''
80 EXIT
```

Hier wordt geen rekening gehouden met eventueel geïnstalleerde routines. Dat is verder ook niet mogelijk want als eenmaal de besturing is overgedragen aan basic dan komen we nooit meer terug in de resetroutine.

Het kan misgaan ...

Zet bijvoorbeeld eens JMP #4FA op adres #4FA en voer een reset uit. U bent gedwongen nog een reset uit te voeren, en nog eens en nog eens ...
De Atom hangt dan extreem stevig. Gebruikers van EPOS/3 kunnen het Atom terminal programma verlaten met CTRL-ALT-END. Daarna het terminal programma opnieuw starten en heel snel ?#4FA=0 intypen voordat u weer een reset geeft.
Immers, EPOS/3 vraagt aan de PC waarom er een reset is geweest. Als dat is omdat het terminal programma gestart is dan wordt alleen de resetroutine in EPROM uitgevoerd. U weet nu waarom.
Gebruikt u echter oudere versies dan blijft er niets anders over dan de pc uit en aan te zetten want u kunt niet ontsnappen met bovengenoemde noodgreep. *PC intypen werkt ook niet. Het resetten van de pc is niet voldoende want in de Atom blijft die hangende JMP staan die u alleen kunt verwijderen door het systeem spanningsloos te maken; uitzetten dus.

Met vriendelijke groeten,

Roland Leurs
Hooge Hoeven 62
4191 MN Geldermalsen

Telefoon: 0345-572970
Fidonet: 2:285/226.9
Aconet: 77:8500/504.32

X M S geheugen

geheugenuitbreiding voor Atom-in-PC tot 4 MB

XMS: hoe en wat?

Tegenwoordig is het een rage om maar veel geheugen in je computer te hebben. Ook voor de Atom wil ik dit verschijnsel introduceren. Alleen hoe ga je dat aanpakken? Iemand die 4 MB wil kopen stopt dat geld (nadat het is omgewisseld in simm) liever in zijn pc dan in een Atom. Maar wat in de pc zit kan door de Atom gebruikt worden is mijn motto. En zo ook het geheugen.

Bij de ontwikkeling van deze drivertjes ben ik uitgegaan van de volgende gedachte: stel ik ga hardwarematig 4 MB toevoegen; hoe doe ik dat? Mijn visie is dat ik dan banken maak van 16 kb die in het geheugengebied #4000-#7FFF geplaatst worden. Met een schakelbyte kan een bank geselecteerd worden, net zoals bij de schakelkaart met uitbreidings-eproms. Dit geheugengebied waar het 16 kb blok in geplaatst wordt noemen we 'paginaframe'.

Uitgaande van deze aanpak zijn de drivers gemaakt. Voor het omschakelen tussen de diverse geheugenbanken is een OSBYTE-call toegevoegd. Het daadwerkelijk omschakelen geschied door het schrijven van de huidige pagina naar de pc en van de pc de nieuwe pagina in te lezen. Deze akties worden dus door de OSBYTE-call uitgevoerd. De programmeur hoeft zich daar niet druk over te maken.

XMS: het gebruik vanuit programma's

De volgende OSBYTE-calls zijn toegevoegd in de XMS-driver:

osbyte F0: versie opvragen
 -invoer: geen
 -uitvoer: A=aantal xms pagina's
  X=versienummer
  Y=releasenummer

osbyte F3: pagina voorzetten
 -invoer: Y=pagina nummer
 -uitvoer: geen

osbyte F4: paginaframe instellen/opvragen
-invoer: Y=hoge byte paginaframe
-uitvoer: Y=hoge byte paginaframe

OSBYTE#F0 geeft informatie omtrent de versie van de XMSDRV.RUN driver. Maar ook het aantal beschikbare XMS pagina's is hiermee op te vragen. Door dit met 16k te vermenigvuldigen weet u hoeveel kb geheugen nog beschikbaar is.

OSBYTE#F3 "schakelt" de opgegeven pagina voor. De inhoud van #4000-#7FFF wordt naar de pc gestuurd en de nieuwe pagina wordt ingelezen en weer in dit geheugengebied geschreven.

OSBYTE#F4 stelt het adres van de 16 kb geheugenbank in. Waarschijnlijk is #4000-#7FFF niet altijd een gelukkige keuze voor het paginaframe. Met deze functie is het paginaframe instelbaar in het gebied #2000-#9FFF. Het begin van het frame ligt dus tussen #2000 en #6000. Deze functie kan ook gebruikt worden om het begin van het paginaframe op te vragen. Geef daartoe een ongeldige waarde (bijv. #00) op als parameter. Na een reset staat het frame op #4000.

De functies #F1 en #F2 zijn ook beschikbaar; ze schakelen ook een nieuwe pagina voor met met de volgende opties:

osbyte F1: pagina voorzetten
-invoer: Y=pagina nummer
-uitvoer: geen
opmerking: bij dit commando wordt de huidige
pagina niet opgeslagen in xmsgeheugen

osbyte F2: pagina voorzetten
-invoer: Y=pagina nummer
-uitvoer: geen
opmerking: bij dit commando wordt de nieuwe
pagina niet gelezen in het atomgeheugen

OSBYTE#F1 schakelt een nieuwe pagina voor maar de huidige pagina wordt niet opgeslagen in het xms geheugen. OSBYTE#F2 schakelt een nieuwe pagina voor zonder deze in de Atom in te lezen (in het pageframe blijven dus de data van de vorige pagina staan).
Deze OSBYTE-calls kunnen gebruikt worden om tijd te sparen. Als u het XMS-geheugen gaat vullen met data is het overbodig om een pagina weg te schrijven, nieuwe data inlezen en die meteen overschrijven met nieuwe data.
Gebruik de calls #F1 en #F2 echter met begrip waar u mee bezig bent.

XMS; de drivers

In de pc draait een TSR (achtergrond)programma dat door de Atom via commando #4B kan worden opgeroepen. Dit programma XMS.COM kan als parameter de grootte van het XMS-geheugen bevatten. Dit dient minimaal 16 kb te zijn. Maximaal wordt 4096 kb gereserveerd. Als het programma zonder parameter wordt opgeroepen zoekt de driver zelf uit hoeveel XMS geheugen beschikbaar is, met een maximum van 256 kb.
Het programma gebruikt ongeveer 17 kb conventioneel geheugen in de pc, inclusief een buffer om 16 kb data tijdelijk op te slaan. Door het programma nogmaals te starten deïnstalleert het zichzelf weer en wordt alle geheugen weer vrij gegeven.

Mensen die met Windows 3.xx werken kunnen het beste een batchfile maken die de XMS driver installeert en daarna het terminal programma opstart. De batchfile wordt afgesloten met een tweede oproep van de XMS driver. Bijv:

```
XMS 4096
ATOM /r=esc
XMS
```

Dit heeft als voordeel dat Windows het XMS geheugen beschikbaar stelt i.p.v. MS-DOS. Als u eerst XMS.COM start met 4096 kb XMS voor de Atom dan kan Windows daarna geen gebruik meer van maken en dat betekent performance verlies.
Onder Windows en OS/2 kunt u gerust 4 MB XMS geheugen reserveren, zelfs als dat fysiek niet aanwezig is in uw pc. Windows en OS/2 plaatsen het Atom XMS geheugen (dat relatief weinig benadert wordt) toch in de swapfile. En het staat leuk op het scherm als gemeld wordt dat er nog 4096 kb geheugen beschikbaar is ...

Aan Atom-zijde wordt de uitbreiding met de OSBYTE-calls geladen. Deze driver, XMSDRV.RUN, maakt gebruik van de reset extention vector op adres #4FA. Daardoor wordt na iedere reset gecontroleerd of XMS.COM in de pc geladen is en zo ja, wordt de OSBYTE-vector omgeleid naar de XMSDRV.RUN routine. Een passende mededeling wordt op het scherm afgebeeld.

Voor het bedieningsgemak kunt u eventueel *XMSDRV opnemen in de INIT-file zodat de driver automatisch geladen wordt bij het initialiseren van de Atom systeemsoftware.

Het is niet verstandig om de XMS driver rechtstreeks aan te roepen via pccom#4b vanwege verschillende redenen:

1) mogelijk wordt de XMS driver opgenomen in een volgende versie van het terminal programma; de aanroep wordt dan gewijzigd
2) wie weet maakt iemand ooit hardwarematig zo'n geheugenuitbreiding.

Als een van bovenstaande wijzigingen ooit worden doorgevoerd hoeft alleen de

XMSDRV voor de Atom aangepast te worden, al uw programma's werken dan nog steeds, zonder wijzigingen.

3) XMS.COM moet echt geladen zijn anders loopt de uitvoer van pccom#4b keihard vast; uit en aanzetten is dan de enige optie.

XMS: de toepassingen

Op dit moment zijn er nog geen toepassingen. Een van de toepassingen die ik zal brengen is een versie van de ItemTracer die de twee datafiles (goed voor zo'n 300 kb aan data) in het XMS geheugen laadt en dan van daar uit de gewenste items gaat zoeken. Naar verwachting gaat dat een stuk sneller omdat nu niet byte voor byte van disk gelezen wordt, maar complete stukken van 16 kb van pc naar Atom gekopieerd worden. Deze versie zal t.z.t. op Henri's BBS verschijnen.

De overige toepassingen mag u zelf schrijven...

De software vindt u vanzelfsprekend op het Unicorn BBS in de Atom filearea: ATOM_XMS.ZIP. Deze zipfile bevat de volgende bestanden:

XMS.ASM	source van de XMS tsr voor de pc
XMS.COM	geassembleerde XMS tsr
XMSDRV.ASM	source van XMS driver voor de Atom (6502 cross assembler nodig)
MACRO.INC	macro's voor cross assembler
XMSDRV.RUN	geassembleerde XMS driver op #C00 - #FFF

Met vriendelijke groeten,

Roland Leurs
Hooge Hoeven 62
4191 MN Geldermalsen

Telefoon: 0345-572970
Fidonet: 2:285/226.9
Aconet: 77:8500/504.32

Voor de liefhebbers nogmaals een complete opsomming van de OSBYTE-functies:

osbyte F0: versie opvragen
-invoer: geen
-uitvoer: A=aantal xms pagina's
X=versienummer
Y=releasenummer

osbyte F1: pagina voorzetten
-invoer: Y=pagina nummer
-uitvoer: geen
opmerking: bij dit commando wordt de huidige
pagina niet opgeslagen in xmsgeheugen

osbyte F2: pagina voorzetten
-invoer: Y=pagina nummer
-uitvoer: geen
opmerking: bij dit commando wordt de nieuwe
pagina niet gelezen in het atomgeheugen

osbyte F3: pagina voorzetten
-invoer: Y=pagina nummer
-uitvoer: geen
opmerking: de huidige pagina wordt opgeslagen in
het xms geheugen en de nieuwe pagina
wordt in het atomgeheugen gelezen

osbyte F4: paginaframe instellen/opvragen
-invoer: Y=hoge byte paginaframe
-uitvoer: Y=hoge byte paginaframe
opmerking: het paginaframe is 16 kb groot en kan liggen tussen
#2000-#9FFF; de parameter voor deze functie ligt in
het bereik #20...#60. Bij een ongeldige parameter wordt
de functie niet uitgevoerd, alleen wordt het huidige
frame teruggemeld. Na een reset staat het frame op #4000.

.

EEN GROOT-LETTER-EN-CIJFER DISPLAY

Goed 10 jaar geleden schreef ik eens een kien-programma : als lid van een zangkoor, moet er gezorgd worden voor kienen op de St.Cecilia-viering van het koor. Aangezien ook toen al een aantal doven en niet-opletters aanwezig waren , vulde ik dat kien-, alias bingo-programma aan met een weergave van de getrokken cijfers op scherm-formaat , toen dus ook een groot-beeld-TV. Dat werkte wel. maar raakte in de kast. Dezer dagen werd ik nu geconfronteerd met een man , die omdat hij nog maar weinig kan zien , zijn tijd wilde vullen met het beoefenen van morse . Een oude ATOM werd beschikbaar gesteld , het programma van Henk Bastings geinstalleerd, en zo kon er driftig geoefend worden met het herkennen van morse-tekens.

Echter een van de aardigheidjes van het programma van Henk is dat het achteraf laat zien welke letters/cijfers er in morse gezonden zijn. En dat nu, kan de goede man niet lezen, dus zijn opname niet verifieren.
Het toepassen van de spraak-chip, zoals gedemonstreerd door Jaack Geene op de laatste landdagen lag voor de hand, maar die dingen zijn niet meer te krijgen.
Dus dan maar grote letters en cijfers gebruiken. oké.
Maar : voor cijfers zaten er in het kienprogramma wel een aantal prgramma-regels ; voor alle letters van het alphabet is dat echter een reuze klus. (zie hierna).

Alternatief 1.
Henk Bastings stelde voor : neem de letters en cijfers zoals opgeslagen in ons operating system en vergroot het beeld van het raster tot schermgrootte door voor ieder "pixel" van dat raster een blok van voldoende grootte op dezelfde relatieve plaats op het scherm te zetten.
Erg aardig, en wellicht ook het meest evenwichtig , maar wat is er precies opgeslagen en hoe doe je dat?.
Van de grafische tekens is wel in het handboek aangegeven dat ze uit 6 pixels zijn opgebouwd.

Alternatief 2.
Maak een display met 20 elementen, die het scherm vult .
(zie volgende pagina).
In feite de manier waarop op een lcd-display letters worden weergegeven, afgeleid van het oorspronkelijke 7-segment-display.
Door afhankelijk van de te presenteren letter of cijfer een van te voren vastgestelde set van elementen aan te sturen worden de diverse letters gevormd .
Bezien we de uitvoering , dan blijkt weer dat een aantal patronen van blokken vaker voorkomen en dus afzonderlijk vastgelegde kunnen worden. (zie voorbeeld).

A 1,2,3,4, 17,18, 19

B 1,2,3,4, 17,19,16, 9,10,11,12

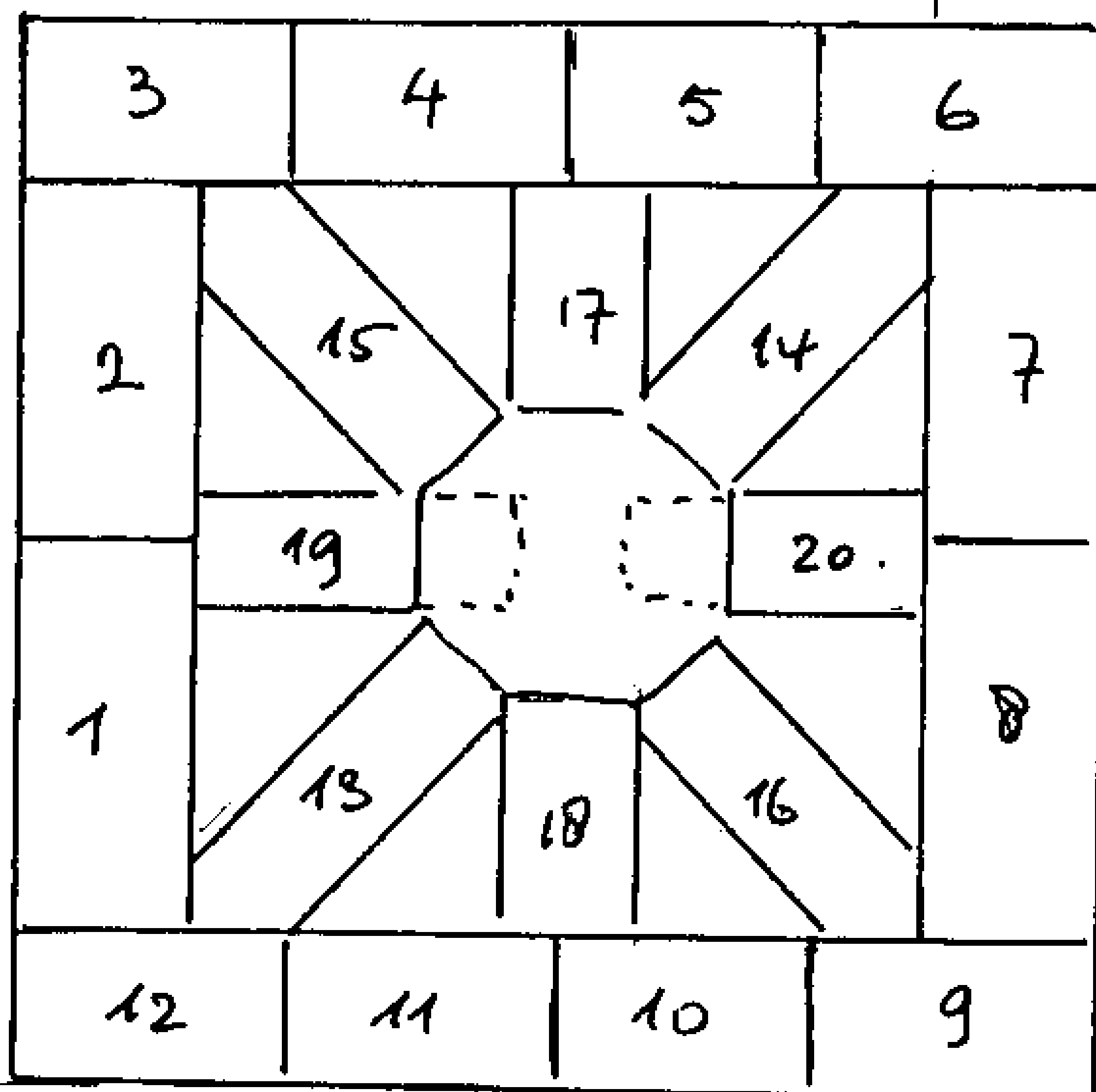
C 1,2,3,4, 5, 9,10,11,12

D 4, 5,6,7,8, 9,10,11,12, 17,18

E 1,2,3,4, 5, 9,10,11,12, 19

F 1,2,3,4, 5, 19

Enz.



Alternatief 3.??
Teken op het scherm in het traject #8000 - 81FF een letter of cijfer : zet die geheugen-inhoud weg op een bepaald adres en copieer afhankelijk van de gewenste letter de hele zaak weer in het traject 8000 81FF ; nog niet aan gerekend , maar we komen plaats tekort.

Alternatief 4. Wie ??.

Gaarne commentaar :
Bruno Tossaint

```
 10REM KIENSPEL
 12REM DOOR B. TOSSAINT MAASTRICHT
 15!#80=#85FFE320;!#84=#608A
 16CLEAR0
 20P.$12;?#E1=0;P."    KIEN PLEZIERIG ....."'''
 30P." DE TREKKING WORDT UITGEVOERD"''
 40P." M.B.V. EEN ACORN ATOM COMPUTER"''
 50P."*************************************************************"
 60P."********                  succes              *"
 70P."********************"
 80P."**********************************************"
 90F.I=0TO50;WAIT;N.
100 GOS.i
110DIMA(91),B(91),C(91)
120F.T=1 TO 91;A?T=T;B?T=0;C?T=0;N.T
130T=1;@=3
140f LINK #80;?#E1=0
150IF ?#8A=66;GOS.a;G.f
160IF ?#8A=71;GOS.b;G.f
170IF ?#8A=86;GOS.c;G.f
180IF ?#8A=72;GOS.i;G.f
190IF ?#8A=78;CLEAR0;P.'"DRUK return VOOR TREKKING";G.120
195IF ?#8A=13;G.200
197G.f
200CLEAR0
210REM KIESROUTINE
220R=A.R.%90+1
230IF A?R=0;G.220
240REM OVERZETTEN GETROKKEN GETAL
250A?R=0;B?R=R;C?T=R
260IF T<=90;G.d
270gP.'"DRUK nIEUW SPEL OF eINDE  ";DOLI.#80;U.?#8A=78OR?#8A=69
280IF ?#8A=78;CLEAR0;G.120
290P.$12''''''''"                einde"'''''''''';E.
300aP.$30;CLEAR0;P.'"   DE TE TREKKEN NUMMERS ZIJN"'
310P."================================="';W=A
320e M=1;L=10
330DO
340F.I=M TO L
350P.W?I
360N.;P.'
370M=M+10;L=L+10
380U.L=100
390P.''"DRUK return VOOR VERVOLG";R.
400bP.$30;CLEAR0;P.'"   DE GETROKKEN NUMMERS ZIJN"'
410P."================================="';W=B
420GOS.e
430R.
440cP.$30;CLEAR0;P.'"  DE VOLGORDE VAN TREKKING IS"'
450P."================================="';W=C
```

```
 460GOS.e
 470R.
 480d Z=R/10;X=R%10
 490CLEAR0
 500IF Z=0;Y=16;G.h
 510Y=0;GOS.(Z*100+1000);Y=32
 520hGOS.(X*100+1000)
 530P.$11'"TREKKING NR."T"   NUMMER  "R
 540IF T=90;P.$11$7$7$7$7$7;F.I=0TO32;P." ";N.;P.$11;G.g
 550T=T+1
 560G.f
 570iP.$30;CLEAR0;P."OM NUMMERS TE CONTROLEREN DRUK :"'
 580P.'" b VOOR DE BEGIN-STAND"
 590P.''" g VOOR GETROKKEN NUMMERS"
 600P.''" v VOOR TREKKING IN VOLGORDE"
 610P.''" n VOOR EEN NIEUW SPEL"
 620P.''" h VAN HELP GEEFT MENU"
 625P.''"GEEF return  VOOR TREKKING"'
 630P.'"    MAAK UW KEUZE :"
 640 RETURN
1000 REM CIJFER 0
1010MOVE(Y+2),42;DRAW(Y+6),45;DRAW(Y+24),45;DRAW(Y+28),42
1020DRAW(Y+28),9;DRAW(Y+24),6;DRAW(Y+6),6;DRAW(Y+2),9
1030DRAW(Y+2),42
1040MOVE(Y+8),36;DRAW(Y+12),39;DRAW(Y+18),39;DRAW(Y+22),36
1050DRAW(Y+22),15;DRAW(Y+18),12;DRAW(Y+12),12;DRAW(Y+8),15
1060DRAW(Y+8),36;R.
1100 REM CIJFER 1
1110MOVE(Y+8),39;DRAW(Y+14),45;DRAW(Y+20),45;DRAW(Y+20),11
1120DRAW(Y+24),11;DRAW(Y+24),6
1130DRAW(Y+10),6;DRAW(Y+10),11;DRAW(Y+14),11
1140DRAW(Y+14),36;DRAW(Y+8),30;DRAW(Y+8),39;R.
1200 REM CIJFER 2
1210MOVE(Y+2),36;DRAW(Y+6),39;DRAW(Y+19),39;DRAW(Y+22),36
1220DRAW(Y+22),32;DRAW(Y+2),12;DRAW(Y+2),6;DRAW(Y+28),6
1230DRAW(Y+28),11;DRAW(Y+11),11;DRAW(Y+28),28
1240DRAW(Y+28),42;DRAW(Y+24),45;DRAW(Y+6),45;DRAW(Y+2),42
1250DRAW(Y+2),36;R.
1300 REM CIJFER 3
1310MOVE(Y+2),36;DRAW(Y+2),45;DRAW(Y+28),45;DRAW(Y+28),29
1320DRAW(Y+22),29;DRAW(Y+20),27;DRAW(Y+20),24;DRAW(Y+22),22
1330DRAW(Y+28),22;DRAW(Y+28),6
1340DRAW(Y+2),6;DRAW(Y+2),15
1350DRAW(Y+6),15;DRAW(Y+6),11;DRAW(Y+24),11
1360DRAW(Y+24),19;DRAW(Y+18),19
1370DRAW(Y+15),22;DRAW(Y+15),29
1380DRAW(Y+18),32;DRAW(Y+24),32;DRAW(Y+24),39
1390DRAW(Y+6),39;DRAW(Y+6),36;DRAW(Y+2),36;R.
1400 REM CIJFER 4
1410MOVE(Y+12),45;DRAW(Y+2),24;DRAW(Y+2),18
1420DRAW(Y+22),18;DRAW(Y+22),6;DRAW(Y+28),6
1430DRAW(Y+28),45;DRAW(Y+22),45;DRAW(Y+22),24;DRAW(Y+8),24
1440DRAW(Y+18),45;DRAW(Y+12),45;R.
```

Een noodkreet van een schrijver

Atom Nieuws verschijnt met de regelmaat van de (seizoens)klok. Daar moest ik aan denken toen ik in het vorige exemplaar van Atom Nieuws, wat nog even terugblikte op het vorige jaar, een woord van dank werd overgebracht aan de schrijvers van de kopij. En op zich is dat natuurlijk heel aardig, zo'n bedankje. Maar aan de andere kant ben ik er eigenlijk niet blij mee. Want, zoals het al zo vaak gaat, staat slechts een enkeling op het podium terwijl er een heel team achter de schermen werkt om tot het uiteindelijk geleverde resultaat te komen.

Daar moest ik even aan denken, bij die opmerking. Want het lijkt zo vanzelfsprekend dat het geschreven materiaal in leesbare en presenteerbare vorm de brievenbus komt bewonen. Maar wie een moment nadenkt over de te leveren inspanning voor een dergelijk nummer, zal toch al gauw tot de conclusie komen dat de Atom-club en met name regio Limburg, een enorme partij werk verzet. Keer op keer. Jaar in jaar uit.

Daarom zou ik als schrijver, en zeker wetend dat de collegae auteurs hier volledig mee instemmen, al diegenen eens in het zonnetje willen zetten, die van al het aangeleverde werk weer een professioneel blad weten te maken. Al het gepuzzel met die oneven aantallen bladen, dan het gepuzzel om alles weer in de juiste volgorde te krijgen. Ik bedoel dit: Sloop het nietje maar eens uit de Atom Nieuws en zie eens hoe een willekeurig A4 velletje moet worden opgebouwd. En dan die kopjes nog steeds keurig boven de onderwerpen, zelfs met de schrijver erbij. Dat is toch weer een heel werk.

Dan het drukken, het stencilen, het vergaren, het nieten, het schrijven van de enveloppen en het likken van de achterkant van Beatrix bij gebrek aan een frankeermachine.

Wie een moment stil staat bij wat werkelijk in stilte gebeurt, zal toch moeten toegeven dat niet alleen de Atom een bijzonder apparaat is, maar ook de Atomclub er eentje is, waar iets bijzonders mee aan de hand is. Want de administratie wordt ook nog gedaan. En iemand beheert het archief. En iemand werkt aan nieuw materiaal voor de Atom kortom: het bruist.

Daarom aan hen die zich met vlijt en ijver inzetten voor de Atomclub inzetten: Bedankt dat we weer een jaar met elkaar AtomClub zijn en AtomNieuws mogen maken!! Leendert

Overzicht leden Federatie Atom Clubs
Nederland-Belgie per 01-04-1996

W.	ARIAANS	JAN v.SALMSTRAAT 61	6121 NE	BORN	
J.J.	BALTUS	B.LUNENSCHLOSZSTR.8	6137 PJ	SITTARD	046-452.10.35
H.	BASTINGS	TERMILESLAAN 113	6229 VT	MAASTRICHT	043-361.54.95
J.K.	BIEL	D.DIJKHUISSTRAAT 56	7558 GB	HENGELO	074-277.00.82
L.	BIJNAGTE	KROMMEKAMP 107	3848 DP	HARDERWIJK	0341-43.08.04
M.D.	BRONS	GERANIUMSTRAAT 40	8013 TL	ZWOLLE	038-421.78.47
D.P.	BRONSDIJK	PAMPUSSTRAAT 21	1435 LG	RYSENHOUT	0297-34.11.27
H.G.	DEKKER	KAARDER 6	1625 TH	HOORN	0229-23.15.15
H.T.G.	DERKSEN	BOLWERK 25	6811 JW	ARNHEM	085-445.54.85
J.	FERON	P.BREUGHELSTRAAT 43	6137 VW	SITTARD	046-451.88.47
J.	GEENE	ZONNEWEIDE 6	5221 BH	DEN BOSCH	073-631.20.80
P.	HARMENS	RIJKSWEG NOORD 310	6136 AJ	SITTARD	046-451.45.06
J.W.	HARTOG	KEYENBERGSEWEG 60	6871 WK	RENKUM	0317-31.37.57
H.v.d.	HEIJDEN	H.v. RANDWIJKSTR. 34	2264 XK	LEIDSCHENDAM	070-327.67.26
E.	HOUNJET	KEEREND 68	6171 VV	STEIN	046-433.26.66
J.	KLEIJN	HUNZE 3	2641 VT	PIJNACKER	
G.W.G.	KROES	INA.B.BAKKERLAAN 117-761	3582 XP	UTRECHT	
J.M.	KROES	POSTBUS 45632	2504 BA	DEN HAAG	
P. van	KUIK	ZUIDEINDE 54 a	1843 JP	GROOT-SCHERMER	0299-67.19.02
J.	LERNOUT	PARKLAAN 4	B-2540	HOVE	003234553473
R.G.M.	LEURS	HOOGE HOEVEN 62	4191 MN	GELDERMALSEN	0345-57.29.70
M.van	LEUVEN	PARCIVALSTRAAT 14	B-2580	St.KATEL.WAVER	003215315082
A. v/d	MEER	HOUTDUIFSTRAAT 43	2623 ER	DELFT	015-262.35.68
E.L.	MEERLOO	TORFLANG 31	7824 XH	EMMEN	
W.van	NORDEN	AURIOLLAAN 3	3527 ER	UTRECHT	030-294.46.85
C.	RUTKOWSKI	Mgr.BUCKXSTR.8	6121 KV	BORN	046-452.80.12
M.H.	RUTTEN	BERKENLAAN 24	3737 RN	GROENEKAN	0346-21.34.95
J.	TEULINGS	KAREL DOORMANSTR.54	5224 GL	DEN BOSCH	073-621.28.88
J.A.	THOLENS	HANOIDREEF 206	3564 HS	UTRECHT	
B.	TOSSAINT	FATIMAPLEIN 85	6214 TW	MAASTRICHT	043-343.16.75
W.	TRUIJEN	HEPPERSTEENWEG 50	B-3680	MAASEIK	003289564792
T.H.	WAAIJER	H.van BOEIJENLAAN 66	2273 DC	VOORBURG	070-386.25.04
P.J. de	WAARD	DORPSTRAAT 125	5504 HD	VELDHOVEN	040-253.34.27
H.	WILLEMARS	LIVINGSTONELAAN 7	3526 HA	UTRECHT	030-289.06.09
C.de	WIT	MUIDERSCHANS 24	3432 XN	NIEUWEGEIN	
J.A. de	WITTE	SCHUINE HONDSBLAAN 58	1852 HR	HEILOO	072-533.47.34
P.A.J.	WOKKE	DUKAATSTRAAT 12	1827 GP	ALKMAAR	072-562.11.35
E.E.P.A.	WORDRAGEN van	SCHIEDAMSEWEG BENEDEN 469b	3028 BM	ROTTERDAM	010-450.94.81

REGIO-ADRESSEN.

Regio NOORD-HOLLAND :
P.v.Kuik, Zuideinde 54-a, 1843 JP Groot Schermer.
tel. 0299-67.19.02

Regio DEN HAAG +ARNHEM
Th.Waayer, H.v.Boeijenlaan 66, 2273 DC Voorburg.
tel. 070-386.25.04

Regio BRABANT-OOST + ZEELAND
J.Teulings, K.Doormansstraat 54, 5224 GL Den Bosch.
tel. 073-621.28.88

Regio LIMBURG + BELGIE EN OOST/NOORD NEDERLAND
C.Rutkowski, Mgr.Buckstr.8 6121 KV Born.
tel. 046-486.01.36

REGIO-MEDEDELINGEN.

1. REGIO BRABANT-OOST

Bijeenkomsten op het bekende adres :
Adolf van Cortenbachstraat 92, Eindhoven, tel. 040-212.32.31
Aanvang 13.30.u

2. REGIO LIMBURG-BELGIE

Clubavonden in "Oos Kaar",Geldersestraat 43, t. 046-432.13.78
op de le vrijdag van de maand.

3. REGIO DEN HAAG

Alles op het nieuwe adres : Theo Waayer
Hendrik v. Boeijenlaan 66, 2273 DC Voorburg,t. 070-386.25.04

4. REGIO ARNHEM e.o.

Geplande bijeenkomsten Acorn Atom Club Regio Arnhem:
Ten Huize van Henri Derksen, Bolwerk 25, 6811 JW ARNHEM
op de derde woensdag van de maand.
Telefoon: 085-445.54.85; UniCorn BBS: 085-442.55.06 xxxx/xxxx
BPS 8N1